UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/053,240	10/26/2001	Louise E. Moser	MOS5082.02A	6071

7590 10/19/2004

John P. O'Banion
O'BANION & RITCHEYLLP
Suite 1550
400 Capitol Mall
Sacramento, CA 95814

EXAMINER	
LOHN, JOSHUA A	

ART UNIT	PAPER NUMBER
2114	

DATE MAILED: 10/19/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/053,240	MOSER ET AL.	
	Examiner	Art Unit	
	Joshua A Lohn	2114	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 May 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-111* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *58-75* is/are allowed.

6)☒ Claim(s) *1-5,7,10-24,26,29-42,44,47-57,76-79,81,84-97,99 and 102-111* is/are rejected.

7)☒ Claim(s) *6,8-9,100,25,27,28,43,45,46,80,82,83,98, and 101* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 October 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *10/26/01*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Claim Objections

Claims 22-23 objected to because of the following informalities: These claims are listed as depending off of claim 1, which would make them duplicates of claims 2 and 3. It is obvious that the applicant's intent if to have these claims depend upon the limitations of claim 21 and will be examined with this interpretation. Appropriate correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-5, 7, 10-24, 26, 29-42, 44, 47-57, 76-79, 81, 84-97, 99, and 102-111 are rejected under 35 U.S.C. 102(e) as being anticipated by Felber et al., United States Patent number 6,574,750, filed January 6, 2000.

As per claim 1, Felber discloses a method for providing fault tolerance between computers of different enterprises across a communication network, comprising: unifying transaction processing and object or process replication between computers across a communication network (Felber, col. 2, lines 17-27); wherein a computer program operating on at least one of said computers can recover from a fault while it is communicating with a program on another of said computers across said communication network (Felber, col. 10, lines 49-56, where communication across network is the accessing of the log used in recovery).

As per claim 2, Felber discloses that the transaction processing protects local data and processing against faults (Felber, col. 2, lines 17-28, where the purpose of replication is to protect against all faults).

As per claim 3, Felber discloses that the replication protects processing and communication across said communication network against faults (Felber, col. 7, lines 20-64, where transaction processing between entities is protected with replication).

As per claim 4, Felber discloses that an object or process operates in a networked mode or a transactional mode (Felber, col. 3, line 55, through col. 4, line 14, where the network mode is the mode in which the client is replicated and the transactional mode is the mode where only the objects of the transaction are replicated).

As per claim 5, Felber discloses wherein in said networked mode, an object or process on one computer can interact with an object or process on an another computer across said communication network (Felber, col. 12, lines 51-52); and wherein an object or process in networked mode is protected against faults by an object or process replication system (Felber, col. 4, lines 5-14).

As per claim 7, Felber discloses wherein an object or process operates in said networked mode by default (Felber, col. 4, lines 5-14, where this is inherent in the client being replicated); and wherein in networked mode an object or process can interact freely with another object or process on the same computer, or with an object or process on another computer across said communication network (Felber, col. 11, line 52, through col. 12, line 52).

As per claim 10, Felber discloses that computers of different enterprises across said communication network maintain consistent views of interactions across said communication network (Felber, col. 10, lines 49-56, where the log maintains consistent views).

As per claim 11, Felber discloses that the messages sent between computers across said communication network are never retracted (Felber, col. 10, lines 5-46, where rollback re-executes, but does not retract messages).

As per claim 12, Felber discloses that a fault in a computer of one enterprise will not abort activities in a computer of another enterprise (Felber, col. 7, lines 20-64, where only the affected objects are rolled back and have activities aborted).

As per claim 13, Felber discloses that the roll-forward recovery is used in networked mode (Felber, col. 4, lines 4-14, and col. 10, lines 5-40, where the use of save points will return to a previous state and any re-execution from a log that occurs constitutes a form of rolling forward); and wherein roll-back/abort recovery is used in transactional mode (Felber, col. 5, lines 21-31, where if only using replicated transaction and no saved states, the system uses roll-back recovery).

As per claim 14, Felber discloses that the roll-forward recovery starts from a checkpoint and then replays messages from a message log (Felber, col. 10, lines 6-36); and wherein messages involved in an aborted transaction are not replayed (Felber, col. 8, lines 40-54, where aborted, or non-committed, transactions are not recorded or logged and thus would not be replayed).

As per claim 15, Felber discloses that roll-forward recovery of one object or process does not disrupt continued operation of another object or process or of a database (Felber, col. 10,

lines 6-56, where the recovery is only for the failed transaction and the database is kept separate from all the transactions to allow continued operation).

As per claim 16, Felber discloses that a message generated during roll-forward recovery of an object or process is detected as a duplicate message and is not processed a second time (Felber, col. 10, lines 6-66, where only those messages necessary for recovery are executed, no message that was previously successful, and not rolled back, will be executed as a duplicate).

As per claim 17, Felber discloses that an object or process recovered using roll-forward recovery receives a reply from another object or process and a value from a database that is the same reply and value received during the initial operation of the object or process (Felber, col. 10, lines 49-56, where the database will provide the stored message, which includes the same values as during initial execution, during the recovery).

As per claim 18, Felber discloses that while an object or process is in transactional mode, a request received from another object or process that is not part of the same transaction is queued until the transaction commits or aborts (Felber, col. 5, lines 9-21, where transaction isolation would inherently require that requests from another transaction not be executed because the objects would no longer be properly isolated).

As per claim 19, Felber discloses that recovery of an object or process restores the state of the object or process and then processes a message that was queued waiting for a transaction to commit or abort (Felber, col. 10, lines 49-56, where the log is used to record all messages to be replayed in recovery, including any pending messages, col. 10, lines 42-45).

As per claim 20, Felber discloses that a message for a current transaction is processed but a message of an enclosing transaction or no transaction remains queued (Felber, col. 5, liens 22-

31, where if the current transaction is a nested transaction, the enclosing transaction may wait for repair of the nested transaction).

As per claim 21, Felber discloses a method for providing fault tolerance between computers of different enterprises across a communication network, comprising: unifying transaction processing and object or process replication between computers across a communication network (Felber, col. 2, lines 17-27); wherein a computer program operating on at least one of said computers can recover from a fault while it is communicating with a program on another of said computers across said communication network (Felber, col. 10, lines 49-56, where communication across network is the accessing of the log used in recovery). Felber also discloses that an object or process operates in a networked mode or a transactional mode (Felber, col. 3, line 55, through col. 4, line 14, where the network mode is the mode in which the client is replicated and the transactional mode is the mode where only the objects of the transaction are replicated).

As per claim 22, Felber discloses that the transaction processing protects local data and processing against faults (Felber, col. 2, lines 17-28, where the purpose of replication is to protect against all faults).

As per claim 23, Felber discloses that the replication protects processing and communication across said communication network against faults (Felber, col. 7, lines 20-64, where transaction processing between entities is protected with replication).

As per claim 24, Felber discloses wherein in said networked mode, an object or process on one computer can interact with an object or process on an another computer across said

communication network (Felber, col. 12, lines 51-52); and wherein an object or process in networked mode is protected against faults by an object or process replication system (Felber, col. 4, lines 5-14).

As per claim 26, Felber discloses wherein an object or process operates in said networked mode by default (Felber, col. 4, lines 5-14, where this is inherent in the client being replicated); and wherein in networked mode an object or process can interact freely with another object or process on the same computer, or with an object or process on another computer across said communication network (Felber, col. 11, line 52, through col. 12, line 52).

As per claim 29, Felber discloses that computers of different enterprises across said communication network maintain consistent views of interactions across said communication network (Felber, col. 10, lines 49-56, where the log maintains consistent views).

As per claim 30, Felber discloses that the messages sent between computers across said communication network are never retracted (Felber, col. 10, lines 5-46, where rollback re-executes, but does not retract messages).

As per claim 31, Felber discloses that a fault in a computer of one enterprise will not abort activities in a computer of another enterprise (Felber, col. 7, lines 20-64, where only the affected objects are rolled back and have activities aborted).

As per claim 32, Felber discloses that the roll-forward recovery is used in networked mode (Felber, col. 4, lines 4-14, and col. 10, lines 5-40, where the use of save points will return to a previous state and any re-execution from a log that occurs constitutes a form of rolling forward); and wherein roll-back/abort recovery is used in transactional mode (Felber, col. 5, lines

21-31, where if only using replicated transaction and no saved states, the system uses roll-back recovery).

As per claim 33, Felber discloses that the roll-forward recovery starts from a checkpoint and then replays messages from a message log (Felber, col. 10, lines 6-36); and wherein messages involved in an aborted transaction are not replayed (Felber, col. 8, lines 40-54, where aborted, or non-committed, transactions are not recorded or logged and thus would not be replayed).

As per claim 34, Felber discloses that roll-forward recovery of one object or process does not disrupt continued operation of another object or process or of a database (Felber, col. 10, lines 6-56, where the recovery is only for the failed transaction and the database is kept separate from all the transactions to allow continued operation).

As per claim 35, Felber discloses that a message generated during roll-forward recovery of an object or process is detected as a duplicate message and is not processed a second time (Felber, col. 10, lines 6-66, where only those messages necessary for recovery are executed, no message that was previously successful, and not rolled back, will be executed as a duplicate).

As per claim 36, Felber discloses that an object or process recovered using roll-forward recovery receives a reply from another object or process and a value from a database that is the same reply and value received during the initial operation of the object or process (Felber, col. 10, lines 49-56, where the database will provide the stored message, which includes the same values as during initial execution, during the recovery).

As per claim 37, Felber discloses that while an object or process is in transactional mode, a request received from another object or process that is not part of the same transaction is

queued until the transaction commits or aborts (Felber, col. 5, lines 9-21, where transaction isolation would inherently require that requests from another transaction not be executed because the objects would no longer be properly isolated).

As per claim 38, Felber discloses that recovery of an object or process restores the state of the object or process and then processes a message that was queued waiting for a transaction to commit or abort (Felber, col. 10, lines 49-56, where the log is used to record all messages to be replayed in recovery, including any pending messages, col. 10, lines 42-45).

As per claim 39, Felber discloses that a message for a current transaction is processed but a message of an enclosing transaction or no transaction remains queued (Felber, col. 5, liens 22-31, where if the current transaction is a nested transaction, the enclosing transaction may wait for repair of the nested transaction).

As per claim 40, Felber discloses a method for providing fault tolerance between computers of different enterprises across a communication network, comprising: unifying transaction processing and object or process replication between computers across a communication network (Felber, col. 2, lines 17-27); wherein a computer program operating on at least one of said computers can recover from a fault while it is communicating with a program on another of said computers across said communication network (Felber, col. 10, lines 49-56, where communication across network is the accessing of the log used in recovery). Felber further discloses that an object or process operates in a networked mode or a transactional mode (Felber, col. 3, line 55, through col. 4, line 14, where the network mode is the mode in which the client is replicated and the transactional mode is the mode where only the objects of the

transaction are replicated). Felber also discloses wherein in said networked mode, an object or process on one computer can interact with an object or process on an another computer across said communication network (Felber, col. 12, lines 51-52); and wherein an object or process in networked mode is protected against faults by an object or process replication system (Felber, col. 4, lines 5-14).

As per claim 41, Felber discloses that the transaction processing protects local data and processing against faults (Felber, col. 2, lines 17-28, where the purpose of replication is to protect against all faults).

As per claim 42, Felber discloses that the replication protects processing and communication across said communication network against faults (Felber, col. 7, lines 20-64, where transaction processing between entities is protected with replication).

As per claim 44, Felber discloses wherein an object or process operates in said networked mode by default (Felber, col. 4, lines 5-14, where this is inherent in the client being replicated); and wherein in networked mode an object or process can interact freely with another object or process on the same computer, or with an object or process on another computer across said communication network (Felber, col. 11, line 52, through col. 12, line 52).

As per claim 47, Felber discloses that computers of different enterprises across said communication network maintain consistent views of interactions across said communication network (Felber, col. 10, lines 49-56, where the log maintains consistent views).

As per claim 48, Felber discloses that the messages sent between computers across said communication network are never retracted (Felber, col. 10, lines 5-46, where rollback re-executes, but does not retract messages).

As per claim 49, Felber discloses that a fault in a computer of one enterprise will not abort activities in a computer of another enterprise (Felber, col. 7, lines 20-64, where only the affected objects are rolled back and have activities aborted).

As per claim 50, Felber discloses that the roll-forward recovery is used in networked mode (Felber, col. 4, lines 4-14, and col. 10, lines 5-40, where the use of save points will return to a previous state and any re-execution from a log that occurs constitutes a form of rolling forward); and wherein roll-back/abort recovery is used in transactional mode (Felber, col. 5, lines 21-31, where if only using replicated transaction and no saved states, the system uses roll-back recovery).

As per claim 51, Felber discloses that the roll-forward recovery starts from a checkpoint and then replays messages from a message log (Felber, col. 10, lines 6-36); and wherein messages involved in an aborted transaction are not replayed (Felber, col. 8, lines 40-54, where aborted, or non-committed, transactions are not recorded or logged and thus would not be replayed).

As per claim 52, Felber discloses that roll-forward recovery of one object or process does not disrupt continued operation of another object or process or of a database (Felber, col. 10, lines 6-56, where the recovery is only for the failed transaction and the database is kept separate from all the transactions to allow continued operation).

As per claim 53, Felber discloses that a message generated during roll-forward recovery of an object or process is detected as a duplicate message and is not processed a second time (Felber, col. 10, lines 6-66, where only those messages necessary for recovery are executed, no message that was previously successful, and not rolled back, will be executed as a duplicate).

As per claim 54, Felber discloses that an object or process recovered using roll-forward recovery receives a reply from another object or process and a value from a database that is the same reply and value received during the initial operation of the object or process (Felber, col. 10, lines 49-56, where the database will provide the stored message, which includes the same values as during initial execution, during the recovery).

As per claim 55, Felber discloses that while an object or process is in transactional mode, a request received from another object or process that is not part of the same transaction is queued until the transaction commits or aborts (Felber, col. 5, lines 9-21, where transaction isolation would inherently require that requests from another transaction not be executed because the objects would no longer be properly isolated).

As per claim 56, Felber discloses that recovery of an object or process restores the state of the object or process and then processes a message that was queued waiting for a transaction to commit or abort (Felber, col. 10, lines 49-56, where the log is used to record all messages to be replayed in recovery, including any pending messages, col. 10, lines 42-45).

As per claim 57, Felber discloses that a message for a current transaction is processed but a message of an enclosing transaction or no transaction remains queued (Felber, col. 5, liens 22-31, where if the current transaction is a nested transaction, the enclosing transaction may wait for repair of the nested transaction).

As per claim 76, Felber discloses a method for providing fault tolerance between computers of different enterprises across a communication network, comprising: unifying transaction processing and object or process replication between computers across a

communication network (Felber, col. 2, lines 17-27); wherein a computer program operating on at least one of said computers can recover from a fault while it is communicating with a program on another of said computers across said communication network (Felber, col. 10, lines 49-56, where communication across network is the accessing of the log used in recovery). Felber further discloses that an object or process operates in a networked mode or a transactional mode (Felber, col. 3, line 55, through col. 4, line 14, where the network mode is the mode in which the client is replicated and the transactional mode is the mode where only the objects of the transaction are replicated). Felber also discloses that the roll-forward recovery is used in networked mode (Felber, col. 4, lines 4-14, and col. 10, lines 5-40, where the use of save points will return to a previous state and any re-execution from a log that occurs constitutes a form of rolling forward); and wherein roll-back/abort recovery is used in transactional mode (Felber, col. 5, lines 21-31, where if only using replicated transaction and no saved states, the system uses roll-back recovery).

As per claim 77, Felber discloses that the transaction processing protects local data and processing against faults (Felber, col. 2, lines 17-28, where the purpose of replication is to protect against all faults).

As per claim 78, Felber discloses that the replication protects processing and communication across said communication network against faults (Felber, col. 7, lines 20-64, where transaction processing between entities is protected with replication).

As per claim 79, Felber discloses wherein in said networked mode, an object or process on one computer can interact with an object or process on an another computer across said communication network (Felber, col. 12, lines 51-52); and wherein an object or process in

networked mode is protected against faults by an object or process replication system (Felber, col. 4, lines 5-14).

As per claim 81, Felber discloses wherein an object or process operates in said networked mode by default (Felber, col. 4, lines 5-14, where this is inherent in the client being replicated); and wherein in networked mode an object or process can interact freely with another object or process on the same computer, or with an object or process on another computer across said communication network (Felber, col. 11, line 52, through col. 12, line 52).

As per claim 84, Felber discloses that computers of different enterprises across said communication network maintain consistent views of interactions across said communication network (Felber, col. 10, lines 49-56, where the log maintains consistent views).

As per claim 85, Felber discloses that the messages sent between computers across said communication network are never retracted (Felber, col. 10, lines 5-46, where rollback re-executes, but does not retract messages).

As per claim 86, Felber discloses that a fault in a computer of one enterprise will not abort activities in a computer of another enterprise (Felber, col. 7, lines 20-64, where only the affected objects are rolled back and have activities aborted).

As per claim 87, Felber discloses that the roll-forward recovery starts from a checkpoint and then replays messages from a message log (Felber, col. 10, lines 6-36); and wherein messages involved in an aborted transaction are not replayed (Felber, col. 8, lines 40-54, where aborted, or non-committed, transactions are not recorded or logged and thus would not be replayed).

As per claim 88, Felber discloses that roll-forward recovery of one object or process does not disrupt continued operation of another object or process or of a database (Felber, col. 10, lines 6-56, where the recovery is only for the failed transaction and the database is kept separate from all the transactions to allow continued operation).

As per claim 89, Felber discloses that a message generated during roll-forward recovery of an object or process is detected as a duplicate message and is not processed a second time (Felber, col. 10, lines 6-66, where only those messages necessary for recovery are executed, no message that was previously successful, and not rolled back, will be executed as a duplicate).

As per claim 90, Felber discloses that an object or process recovered using roll-forward recovery receives a reply from another object or process and a value from a database that is the same reply and value received during the initial operation of the object or process (Felber, col. 10, lines 49-56, where the database will provide the stored message, which includes the same values as during initial execution, during the recovery).

As per claim 91, Felber discloses that while an object or process is in transactional mode, a request received from another object or process that is not part of the same transaction is queued until the transaction commits or aborts (Felber, col. 5, lines 9-21, where transaction isolation would inherently require that requests from another transaction not be executed because the objects would no longer be properly isolated).

As per claim 92, Felber discloses that recovery of an object or process restores the state of the object or process and then processes a message that was queued waiting for a transaction to commit or abort (Felber, col. 10, lines 49-56, where the log is used to record all messages to be replayed in recovery, including any pending messages, col. 10, lines 42-45).

As per claim 93, Felber discloses that a message for a current transaction is processed but a message of an enclosing transaction or no transaction remains queued (Felber, col. 5, liens 22-31, where if the current transaction is a nested transaction, the enclosing transaction may wait for repair of the nested transaction).

As per claim 94, Felber discloses a method for providing fault tolerance between computers of different enterprises across a communication network, comprising: unifying transaction processing and object or process replication between computers across a communication network (Felber, col. 2, lines 17-27); wherein a computer program operating on at least one of said computers can recover from a fault while it is communicating with a program on another of said computers across said communication network (Felber, col. 10, lines 49-56, where communication across network is the accessing of the log used in recovery). Felber also discloses that an object or process operates in a networked mode or a transactional mode (Felber, col. 3, line 55, through col. 4, line 14, where the network mode is the mode in which the client is replicated and the transactional mode is the mode where only the objects of the transaction are replicated). Felber further discloses that while an object or process is in transactional mode, a request received from another object or process that is not part of the same transaction is queued until the transaction commits or aborts (Felber, col. 5, lines 9-21, where transaction isolation would inherently require that requests from another transaction not be executed because the objects would no longer be properly isolated).

As per claim 95, Felber discloses that the transaction processing protects local data and processing against faults (Felber, col. 2, lines 17-28, where the purpose of replication is to protect against all faults).

As per claim 96, Felber discloses that the replication protects processing and communication across said communication network against faults (Felber, col. 7, lines 20-64, where transaction processing between entities is protected with replication).

As per claim 97, Felber discloses wherein in said networked mode, an object or process on one computer can interact with an object or process on an another computer across said communication network (Felber, col. 12, lines 51-52); and wherein an object or process in networked mode is protected against faults by an object or process replication system (Felber, col. 4, lines 5-14).

As per claim 99, Felber discloses wherein an object or process operates in said networked mode by default (Felber, col. 4, lines 5-14, where this is inherent in the client being replicated); and wherein in networked mode an object or process can interact freely with another object or process on the same computer, or with an object or process on another computer across said communication network (Felber, col. 11, line 52, through col. 12, line 52).

As per claim 102, Felber discloses that computers of different enterprises across said communication network maintain consistent views of interactions across said communication network (Felber, col. 10, lines 49-56, where the log maintains consistent views).

As per claim 103, Felber discloses that the messages sent between computers across said communication network are never retracted (Felber, col. 10, lines 5-46, where rollback re-executes, but does not retract messages).

As per claim 104, Felber discloses that a fault in a computer of one enterprise will not abort activities in a computer of another enterprise (Felber, col. 7, lines 20-64, where only the affected objects are rolled back and have activities aborted).

As per claim 105, Felber discloses that the roll-forward recovery is used in networked mode (Felber, col. 4, lines 4-14, and col. 10, lines 5-40, where the use of save points will return to a previous state and any re-execution from a log that occurs constitutes a form of rolling forward); and wherein roll-back/abort recovery is used in transactional mode (Felber, col. 5, lines 21-31, where if only using replicated transaction and no saved states, the system uses roll-back recovery).

As per claim 106, Felber discloses that the roll-forward recovery starts from a checkpoint and then replays messages from a message log (Felber, col. 10, lines 6-36); and wherein messages involved in an aborted transaction are not replayed (Felber, col. 8, lines 40-54, where aborted, or non-committed, transactions are not recorded or logged and thus would not be replayed).

As per claim 107, Felber discloses that roll-forward recovery of one object or process does not disrupt continued operation of another object or process or of a database (Felber, col. 10, lines 6-56, where the recovery is only for the failed transaction and the database is kept separate from all the transactions to allow continued operation).

As per claim 108, Felber discloses that a message generated during roll-forward recovery of an object or process is detected as a duplicate message and is not processed a second time (Felber, col. 10, lines 6-66, where only those messages necessary for recovery are executed, no message that was previously successful, and not rolled back, will be executed as a duplicate).

As per claim 109, Felber discloses that an object or process recovered using roll-forward recovery receives a reply from another object or process and a value from a database that is the same reply and value received during the initial operation of the object or process (Felber, col. 10, lines 49-56, where the database will provide the stored message, which includes the same values as during initial execution, during the recovery).

As per claim 110, Felber discloses that recovery of an object or process restores the state of the object or process and then processes a message that was queued waiting for a transaction to commit or abort (Felber, col. 10, lines 49-56, where the log is used to record all messages to be replayed in recovery, including any pending messages, col. 10, lines 42-45).

As per claim 111, Felber discloses that a message for a current transaction is processed but a message of an enclosing transaction or no transaction remains queued (Felber, col. 5, liens 22-31, where if the current transaction is a nested transaction, the enclosing transaction may wait for repair of the nested transaction).

Allowable Subject Matter

Claims 6, 8, 9, 25, 27, 28, 43, 45, 46, 80, 82, 83, 98, 100, and 101 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Claims 58-75 are allowed.

The following is an examiner's statement of reasons for allowance: Claims 58-75 are allowed for the inclusion, within the context of the entirety of the claim limitations, of the limitations of "wherein a computer program operating on at least one of said computers can

recover from a fault while it is communication with a program on another of said computers across said communication network; wherein an object or process operates in a networked mode or a transactional mode; wherein in said transactional mode, an object or process on one computer can interact with an object or process sin a local database, but not with an object or process on another computer across the communication network".

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure is provided on included form PTO-892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joshua A Lohn whose telephone number is (571) 272-3661. The examiner can normally be reached on M-F 8-4.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert Beausoliel can be reached on (571) 272-3645. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JAL

SCOTT BADERMAN
PRIMARY EXAMINER